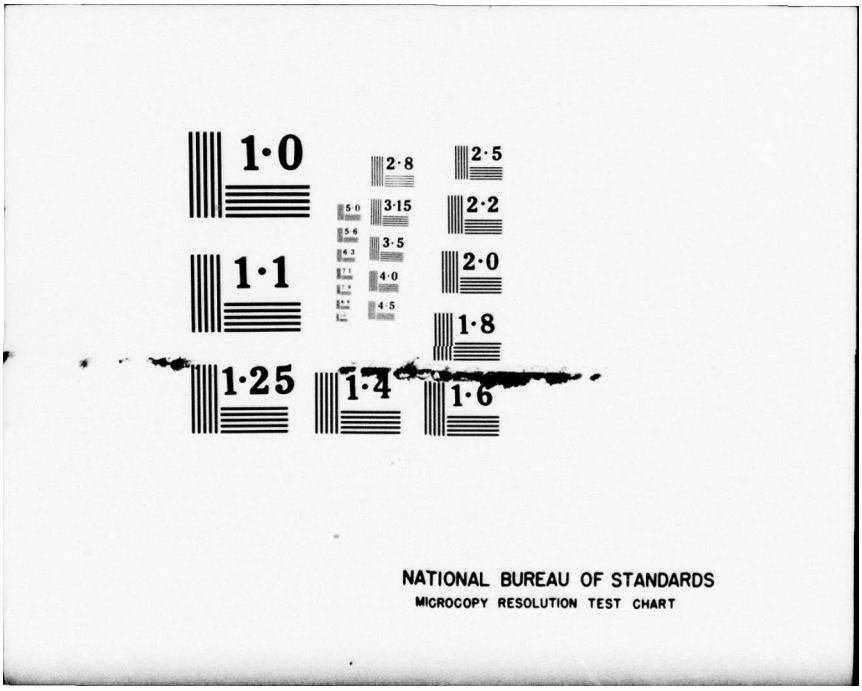
AD-A039 226 AIR FORCE FLIGHT DYNAMICS LAB WRIGHT-PATTERSON AFB OHIO F/G 12/1
SOLVING THE REAL GENERALIZED SYMMETRIC EIGENPROBLEM BY SIMULTAN--ETC(U)
DEC 76 P J NIKOLAI

UNCLASSIFIED AFFDL-TR-76-118 NL

| OF |
ADA
039226

END

DATE
FILMED
6-77


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
6·3
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AFFDL-TR-76-118

(12)

B.S.

SOLVING THE REAL GENERALIZED SYMMETRIC EIGENPROBLEM BY SIMULTANEOUS ITERATION

ANALYSIS AND OPTIMIZATION BRANCH
STRUCTURAL MECHANICS DIVISION

DECEMBER 1976

TECHNICAL REPORT AFFDL-TR-76-118
INTERIM REPORT 1 OCTOBER 1972 - 31 JULY 1976

Approved for public release; distribution unlimited

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (IO) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Paul J. Nikolai

PAUL J. NIKOLAI
Project Scientist

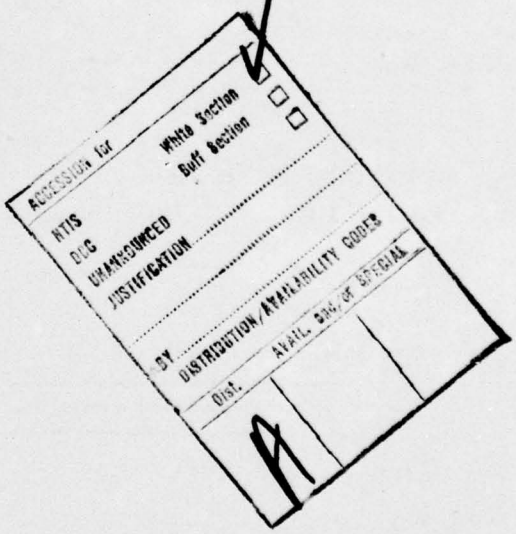
Max Duggins

LtCol Max Duggins
Supervisor

FOR THE COMMANDER:

Howard L. Farmer

HOWARD L. FARMER, Colonel, USAF
Chief, Structural Mechanics Division



Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE - 4 APRIL 77 - 500

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFFDL-TR-76-118 ✓	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Solving the Real Generalized Symmetric Eigenproblem by Simultaneous Iteration	5. TYPE OF REPORT & PERIOD COVERED: Technical - Interim 1 Oct 72 - 31 Jul 76
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): Paul J. Nikolai	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Flight Dynamics Laboratory (AFFDL/FBR) ✓ Air Force Wright Aeronautical Laboratories AFSC, Wright-Patterson Air Force Base, OH 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: DOD Element 61102F 70710103
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Flight Dynamics Laboratory (AFFDL/FBR) Air Force Wright Aeronautical Laboratories AFSC, Wright-Patterson Air Force Base, OH 45433	12. REPORT DATE: December 1976
	13. NUMBER OF PAGES: 32
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

012 070

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

eigenvalues
eigenvectors
diagonable matrices
simultaneous iteration
sparse matrices
numerical linear algebra
computer software

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report provides a USA Standard FORTRAN implementation of the simultaneous iteration algorithm for partial solution of the eigenvalue problem for a real matrix symmetric relative to a prescribed inner product. Also included is a brief discussion of the operation of the algorithm together with the solution of a sample problem executed using FORTRAN Extended, Version 4, under the NOS/BE operating system for the CDC 6600.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This report describes work performed in the Aerospace Research Laboratories and in the Air Force Flight Dynamics Laboratory under the Defense Research Sciences Program, Project 7071, Research in Applied Mathematics, Task 01, Computational Aspects of Fluid and Structural Mechanics. The work described was carried out between October 1972 and July 1976. This is an interim report. Further reports in this series will be documented under Project 2304, Mathematical and Information Sciences, managed by the Air Force Office of Scientific Research, Bolling AFB, DC.

TABLE OF CONTENTS

SECTION I

INTRODUCTION

The structural analysis of mathematical models of complex aerospace and naval vehicles often involves the partial solution of the generalized algebraic eigenvalue problem. A pair of n-square matrices A and B are given and it is required to calculate one or more non-zero eigenvectors u and corresponding complex eigenvalues λ for which $Au = \lambda Bu$. In many familiar applications the matrices A and B are real and symmetric and B is positive definite so that all arithmetic is real. But n is very large. Often A and B have relatively few non-zero entries; i.e., A and B are sparse. Nevertheless, the partial numerical solution of the large generalized symmetric eigenproblem is a challenge to the numerical analyst and the systems programmer alike. The terms "large" and "very large" are used here relative to the computer at hand and the limitations imposed by its operating system. One may distinguish three cases. If at most one copy of A and B can be stored in the central memory of the machine, we say that n is large. If n is so great that only a relatively few n-vectors may be simultaneously stored, then n is very large. If at most one n-vector can be accomodated, then n is simply gigantic.

This report describes the simultaneous iteration method for the partial solution of the generalized symmetric eigenproblem and gives a USA Standard FORTRAN implementation of the algorithm called SUBROUTINE SIMITZ. The use of SIMITZ is indicated if n is large or very large,

but it will not handle gigantic problems. However, it may be appropriate to problems of moderate size when only a few eigenvalues or eigenvectors are required.

Section II of this report provides a brief mathematical formulation of the simultaneous iteration method and describes the Rutishauser-Reinsch algorithm *ritzit* upon the program SIMITZ is based. Section III is devoted to a discussion of alternative methods for solution of the large eigenproblem and their comparison with simultaneous iteration. Detailed instructions for use of SIMITZ are given in Section IV and a complete listing of the FORTRAN program comprises Section V.

SECTION II

DESCRIPTION

The present program is an implementation of the simultaneous iteration algorithm [2] for calculating the eigenvalues largest in magnitude and corresponding eigenvectors of a real matrix symmetric relative to a prescribed inner product. Let ip(n, w, z) denote an inner product in the space of real column n-tuples and let the real n-square matrix C satisfy ip(n, Cw, z) = ip(n, w, Cz). Then C is symmetric relative to ip, and if the n-square positive definite matrix B satisfies ip(n, w, z) = $w^T Bz$ then C is B-symmetric. The equation BC = $C^T B$ characterizes the B-symmetry of C. Given an optional set of p initial approximate eigenvectors of a real n-square B-symmetric matrix C corresponding to p eigenvalues of C largest in magnitude, the program calculates em eigenvalues and em corresponding eigenvectors, $0 \leq em < p \leq n$, to a precision dependent on the structure of C and on a prescribed tolerance eps. The matrix B is presented to the program as an independently prepared real function subprogram which calculates ip(n, w, z) = $w^T Bz$ given column n-vectors w and z. The matrix C is presented as an independently prepared subroutine subprogram op(n, w, z) which when given an n-vector z computes its image w = Cz. The program is an outgrowth of a literal FORTRAN translation [6] of the ALGOL procedure *ritzit* [9] to which it is substantially equivalent when ip(n, w, z) = $w^T z$, the standard inner product. But depending on the choice of B and C, the present

program enables the direct treatment of a wide variety of symmetric eigenproblems.

Let $A = A^T$ and $B = B^T$ denote n-square real matrices and let σ be real. If B is positive definite then the matrix $C = B^{-1}(A - \sigma B)$ is B-symmetric, and the program computes eigenvalues farthest from σ of the eigenproblem $Au = \lambda Bu$ and corresponding eigenvectors. Implementation of op(n, w, z) here consists in providing for the appropriate solution for w of the linear system $Bw = (A - \sigma B)z$. Alternatively, selection of op to solve the system $(A - \sigma B)w = Bz$ for w enables the calculation by simultaneous inverse iteration of the eigenvalues nearest to σ and their eigenvectors. Implications for large sparse systems for which the Cholesky factorization [7] of B is impractical are clear.

Let the eigenvalues $d_1, \ldots, d_p, d_{p+1}, \ldots, d_n$ of C be arranged in order of descending absolute value and let E_p denote the direct sum of the distinct eigenspaces corresponding to $d_1, \ldots, d_p$. Let X_0 denote an n-by-p matrix having a p-dimensional column space not orthogonal relative to ip to any eigenvector in E_p. Simultaneous iteration is based on the observation that if $|d_p| > |d_{p+1}|$, the columns of the matrix $X_{k+m} = C^m X_k$ tend to a basis of E_p as ks = k + m increases. But in practice all of the columns of X_{ks} tend toward the eigenspace E_1 causing loss of information concerning the residual eigenvectors. To counter this tendency, set

$$X_{k+m} = C^m X_k B_{k+m}^{-1} \qquad (1)$$

where the p-square upper triangular matrix B_{ks} is constructed together with X_{ks} by the Gram-Schmidt process to render the columns of X_{ks} orthonormal relative to ip. Now the i-th column vector of X_{ks} converges to the i-th eigenvector of C at a rate proportional to $\max_{2 \leq i \leq p} (|d_i/d_{i-1}|, |d_{i+1}/d_i|)$. Clearly this convergence will be delayed in the presence of eigenvalue clustering. But if $|d_p| - |d_{p+1}|$ is not too small, the column space of X_{ks} will contain a good approximation to the i-th eigenvector even when ks is small.

In order to recover this approximation, a modified Rayleigh-Ritz process is employed. Let Q_{ks} denote an orthogonal matrix which diagonalizes the p-square symmetric matrix $B_{ks}B_{ks}^T$. Then the i-th column vector of

$$X_{k+1} = CX_k B_{k+1}^{-1} Q_{k+1} \tag{2}$$

converges to the i-th eigenvector of C at a rate proportional to $|d_{p+1}/d_i|$ while the entries of the diagonal matrix computed with Q_{ks} and properly ordered offer close approximations to $d_1^2, \ldots, d_p^2$. The true signed eigenvalues need only be computed at termination by diagonalizing the leading (p - 1)-square principal submatrix of $X_{ks}^T BCX_{ks}$, the eigenproblem for C projected on E_{p-1} relative to ip.

The program determines a strategy for employing the devices (1) and (2) based on the distribution of the leading p eigenvalues of C upon which the convergence rate ultimately depends. The selection of values m in (1) is particularly important in this regard in that $C^m X_k$ is replaced by the m-th Chebychev polynomial on an appropriate inter-

val [-e, e] evaluated by a special 3-term recurrence relation and permitting accelerated convergence when values of m are continually large. As a result the convergence quotient lies between $|d_p/d_{em}|$ and exp(-arc cosh $|d_{em}/d_p|$). It is nearer to the first value if $|d_1/d_{em}|$ is large and nearer to the second if the latter quotient is close to one.

As the iteration proceeds through a maximum of |km| iteration steps - km is a program parameter - acceptance tests for the eigenvalues and eigenvectors are conducted following each of the Rayleigh-Ritz steps (2). As soon as the relative increase of $|d_{h+1}|$ is smaller than eps/10, then d_{h+1} is accepted and h, the number of previously accepted eigenvalues, is increased by one. Eigenvectors are accepted in groups of one or more corresponding to clusters of accepted eigenvalues nearly equal in magnitude. If g eigenvectors have already been accepted, let $d_{g+1}, \ldots, d_\ell$ denote such a cluster. For all j, $g + 1 \leq j \leq \ell$, denote by y_j the projection relative to ip of the image Cx_j of the j-th column x_j of X_{ks} on the linear closure of $x_1, \ldots, x_\ell$. Set $f_i = \max_j ||Cx_j - y_j|| / ||Cx_j||$ for $i = g + 1, \ldots, \ell$ where the indicated norm is the Euclidean norm or 2-norm relative to ip. If $|d_\ell| f_\ell/(|d_\ell| - e)$ is smaller than eps then all the x_j, $j = g + 1, \ldots, \ell$, are accepted as eigenvectors and g is increased to ℓ. The error quantities f_i are systematically discounted in accordance with the convergence properties of the algorithm to permit convergence in the presence of excessive round-off error or in case the parameter eps is prescribed unrealistically small. Having determined g eigen-

vectors, the iteration continues with p - g remaining columns of X_{ks} until either em eigenvectors have been calculated or |km| has been exceeded. The program may reduce em if it detects either no progress in convergence of eigenvectors corresponding to smaller eigenvalues or lack of stability in the behavior of larger eigenvalues.

The user may wish to supplement the forgoing outline of the operation of the program by consulting the description of the ALGOL procedure *ritzit* in [9] or [12] as well as a review of the mathematical foundations of simultaneous iteration in [5] and [8]. For this reason we describe the principal differences between *ritzit* and the present program. (a) The procedure *inprod* for calculating standard inner products was removed and the procedure ip was introduced where appropriate. (b) The procedure *jacobi* for calculating the solutions of the eigenproblem for the p-square and (p - 1)-square symmetric matrices was replaced by calls to the EISPACK [9] subroutines TRED2 and IMTQL2, primarily to save space. (c) The procedure *random* for calculating random column n-vectors of the matrix X_{ks} was replaced by in-line code which references a FORTRAN function RANF. RANF returns uniformly distributed random REAL values from the interval (0, 1), one per function reference, given any one argument of any type. It is provided by the user. (d) The procedure *orthog* to perform Gram-Schmidt orthogonalization of the columns of X_{ks} was replaced by internally linked in-line code. In attempting to control potential underflow within *orthog* in a machine independent fashion,

ritzit calculates the machine precision mc but assumes in usage that out-of-range values underflow gracefully to zero, a machine dependent characteristic. The present program utilizes a single REAL machine dependent constant MT, the ratio of the smallest FORTRAN representable positive value to the machine precision, to test for this condition and upon its detection to take appropriate measures. (e) In its ALGOL implementation *ritzit* requires approximately $(p + 3)n + 2p^2 + 5p$ storage locations in excess of those required by the program. Economies resulting in part from (b) above [3] have reduced this requirement to $(p + 2)n + p^2 + 4p$ in the present program. All working storage is confined to a single array of $2n + p^2 + 3p$ locations. (f) The value of km as an input parameter, set to |km| during program execution, is finally replaced by the value of ks as an output parameter, the number of iteration steps used in the calculation of em eigenvectors. (g) The present program retains unchanged the reference to a user supplied procedure *inf* as a window on program execution. However, the one variable involving eps is periodically redefined to enable effective control of eps from *inf* or from ip or op should this prove desirable.

The testing procedures developed for the present FORTRAN program parallel its evolution from a research tool, which conformed closely to its ALGOL parent, to present form. Early testing was concentrated on duplicating the tests furnished with *ritzit* and in eliminating errors in interpretation and translation of the ALGOL code. This was done for the most part on the CDC 6600 using FORTRAN Extended, Ver-

sion 3, under the SCOPE 3.3 operating system. Upon completion of this first phase, the resulting program was distributed as SUBROUTINE RITZIT with a locally developed library of FORTRAN linear algebra routines [6]. This same program served as a basis of *ritzit* translations for IBM 360/370 processors [1,3] whose preparation uncovered several bugs in the RITZIT code and suggested worthwhile modifications. A second phase of testing involved the development of a package of auxiliary FORTRAN programs for use with SUBROUTINE RITZIT to solve the eigenvalue problem $Au = \lambda Bu$ through methods depending on Cholesky factorization where A and B may either be full matrices or sparse and banded. This phase was conducted on the CDC 6600 using FORTRAN Extended, Version 4, under SCOPE 3.4.

Systematic testing of the present program, SUBROUTINE SIMITZ, has been accomplished in part with the aid of driver program TESTB which generates a symmetric band matrix A and a lower triangular band matrix T of prescribed order n and bandwidths whose relevant entries are randomly generated integer values from a prescribed interval. The band matrix B is TT^T, and the program calculates the maximal eigenvalues of $Au = \lambda Bu$. For the sequence of values of p, $p = 2, \ldots, min([n/5], 10)$, TESTB exercises SIMITZ for successive values of em, $em = 1, \ldots, p - 1$. For each value of i, $i = 1, \ldots, em$, TESTB computes the residuals $Ax_i - d_i Bx_i$ and their Euclidean norms relative to the standard inner product. Each norm is normalized by the difference $|d_1| - e$, and for each value of em the quantity $\max_{1 \leq i \leq em} ||Ax_i - d_i Bx_i||/(|d_1| - e)$, the value k of i for which

the maximum occurs, and the corresponding geometric mean with unit weights are listed. Also listed are the relevant non-zero diagonals of A and T and the final eigenvalues computed for em = p - 1.

Figure 1 shows an output listing from the executable program TESTB on the CDC 6600 under the NOS/BE operating system and FORTRAN Extended, Version 4. Here A and B are of order 30 and each of bandwidth 7 having relevant entries between -99 and +99. Listed are the main diagonal and the three adjacent lower diagonals of T and A beginning with the entries in the first column. Here eps = 10^{-10} and km = 100. Note how the relative nearness in magnitude of the first three eigenvalues inhibits the convergence of the second eigenvector when p = 3 and em = 2 resulting in acceptance of the first eigenvector only. The fourth eigenvalue, however, is in absolute value far enough away from this cluster to permit successful convergence when p = 4 and em = 1, 2, and 3. This phenomenon points to a procedure for pursuing a solution when p is initially chosen too small. SIMITZ may be reentered with X containing the approximate eigenvectors calculated for the smaller value of p as initial approximations for use with p increased in size. Significant processor time may often be saved in this fashion.

ADJACENT NON-ZERO LOWER DIAGONALS OF T, TT' = B

53	86	78	60	85	99	73	64	80	88	93	93	57	52	83
74	99	88	87	52	64	86	95	66	86	55	95	61	76	70
-26	22	49	65	-30	38	82	-9	-87	56	18	35	33	-46	-31
1	90	-44	75	-6	-49	99	90	-62	16	2	-40	-67	14	
-25	6	54	32	-73	-35	-1	-46	6	-13	-77	77	-51	-85	31
-24	88	-66	-18	-41	5	80	74	-80	-2	-6	75	-97		
-76	19	-26	-5	97	70	6	-95	45	-30	-12	-55	-92	9	39
97	-19	26	86	-48	-52	-31	49	-8	-85	5	2.			

ADJACENT NON-ZERO LOWER DIAGONALS OF A = A'

-3	50	-55	-73	-47	88	-33	-28	-77	-40	96	78	3	-54	-39
10	95	-53	-80	3	52	44	1	-5.	-94	-22	39	-36	-4	36
0	65	59	-29	-84	-4	41	-91	72	-65	74	36	82	-86	-5
22	56	94	9	-47	84	-97	-1	57	-22	-1	-55	-15	51	
52	5	-41	-95	-47	-91	14	84	-83	69	43	-27	61	-79	14
21	85	-5	-61	-16	-15	90	-44	-22	-66	50	-34	-37		
-17	-94	78	-95	-89	-31	-43	52	-64	-70	71	95	-63	-12	87
91	-93	6	39	66	32	-21	93	-11	8	-98	-12			

P	EM(IN)	EM(OUT)	KS	K	MAX RESIDUAL	MEAN RESIDUAL	TIME(SECS)
2	1	1	30	1	3.70E-11	3.70E-11	.42
3	1	1	26	1	2.81E-11	2.81E-11	.62
3	2	1	56	1	3.00E-11	3.00E-11	1.00
4	1	1	15	1	2.17E-11	2.17E-11	.69
4	2	2	15	2	2.01E-11	1.81E-11	.69
4	3	3	24	3	2.53E-11	2.09E-11	.92
5	1	1	13	1	2.20E-11	2.20E-11	1.02
5	2	2	13	1	2.03E-11	1.99E-11	1.03
5	3	3	13	1	2.24E-11	1.50E-11	1.01
5	4	4	33	1	2.37E-11	6.67E-12	1.44
6	1	1	13	1	2.09E-11	2.09E-11	1.36
6	2	2	13	2	2.16E-11	2.04E-11	1.36
6	3	3	13	2	2.89E-11	2.03E-11	1.33
6	4	4	21	4	2.02E-10	3.63E-11	1.76
6	5	4	31	1	2.64E-11	1.09E-11	2.10

FINAL EIGENVALUES

2.393019487013E+00 -1.245936323012E+00 1.188249083996E+00 -2.273759446561E-01 -9.755629622556E-02

Figure 1. Output Listing

SECTION III

SOME ALTERNATIVE METHODS

The generalized symmetric eigenvalue problem $Au = \lambda Bu$ may in theory be reduced to a standard symmetric eigenvalue problem $Pv = \lambda v$. When B is positive definite a lower triangular matrix T may be calculated by Cholesky's method which satisfies $B = TT^T$. Setting $v = T^T u$ we have $P = T^{-1} A (T^{-1})^T$. This calculation of T may be accomplished even when n is large. But in practice the procedure is complicated by the fact that P need not be sparse even when A and B are so. But $w = Pz$ can be calculated directly from A and T without inversion of T. SIMITZ combined with Cholesky's decomposition eliminates the need for a special inner product - use ip (n, w, z) = $w^T z$ - and so improves efficiency for large n or for n of moderate size. The so called direct methods of treating $Au = \lambda Bu$ which rely on orthogonal similarity transformations to reduce P to tridiagonal form followed by application of the QL or QR algorithm [12] are generally unsuitable for large problems unless the structure of P is quite special. So called iterative methods exemplified by simultaneous iteration must therefore be exploited when n is large.

One of the most popular of such methods is the inverse power method or inverse iteration [12]. Given an approximate eigenvalue σ and an approximate corresponding eigenvector X_0 the iteration $(A - \sigma B) X_k = BX_{k-1}$, $k = 1, \ldots$ converges rapidly to an eigenvector of $Au = \lambda Bu$ when at each stage $X_k^T B X_k = 1$. This resembles the simultaneous iteration algorithm but when p = em = 1 and when the eigen-

value nearest to σ is sought with its eigenvector. The approximate eigenvalue must be calculated first, however, and inverse iteration may fail if σ tends to identify a multiple eigenvalue. Eigenvalues may be obtained by any of the various bisection methods [4,12] or by Rayleigh-Ritz techniques. The inverse power method and its variants are the only methods applicable to gigantic problems known to this writer.

One of the most promising methods for very large problems is the block Lanczos algorithm [11] for which a FORTRAN program is available to solve the standard symmetric eigenvalue problem $Pv = \lambda v$. Block Lanczos has been compared to *ritzit* and found to be superior in many situations. We see no inherent impediment to modifying block Lanczos to produce solutions of $Au = \lambda Bu$ in the manner in which we have modified *ritzit*.

SECTION IV

INSTRUCTIONS FOR THE USER

A FORTRAN executable program or subprogram possessing the control statements equivalent to

```
DIMENSION X(MN,P), D(P), WK(K)
INTEGER P, EM
REAL IP
EXTERNAL IP, INF, OP
```

may call SUBROUTINE SIMITZ into execution via the statement

```
CALL SIMITZ (N, P, KM, EPS, IP, OP, INF, EM, X, MN, D, WK)
```

where

N is an INTEGER input variable, the order of the matrix C.

P is an INTEGER input variable, the number of simultaneous iteration vectors.

KM as an INTEGER input variable is in magnitude the maximum number of iteration steps to be executed. If KM identifies a negative value then P initial approximate eigenvectors are assumed to be present in the array X. Otherwise SIMITZ supplies random initial eigenvectors.

KM as an INTEGER output variable identifies the number KS of iteration steps finally used in the calculation of EM eigenvectors.

EPS is a REAL input variable, the tolerance for accepting eigenvectors.

IP is an EXTERNAL input variable, the name of a FORTRAN compatible REAL FUNCTION subprogram of the form IP(N, Z, W) which must return the inner product (W, BZ) = $W^T BZ$ of the vectors identified by the N-arrays Z and W relative to the positive definite matrix B.

OP is an EXTERNAL input variable, the name of a FORTRAN compatible SUBROUTINE subprogram of the form OP(N, Z, W) which must calculate the image W of the vector identified by the N-array Z under the N-square matrix C without overwriting Z.

INF is an EXTERNAL input variable, the name of a FORTRAN compatible SUBROUTINE subprogram which may be used for obtaining information or to exert control during execution of SIMITZ. INF has the form INF(KS, G, H, F) where

KS is an INTEGER output variable, the number of the next iteration step.

G is an INTEGER output variable, the number of already accepted eigenvectors.

H is an INTEGER output variable, the number of already accepted eigenvalues.

F is a REAL output variable P-array, error quantities measuring respectively the state of convergence of the P simultaneous iteration vectors. In addition, if convergence fails in SUBROUTINE IMTQL2 after G eigenvectors have been accepted, then F(G+1) is replaced by 1000.*FLOAT (IERR) where IERR is the error

indicator output by IMTQL2. Each element of the array F is initially set by SIMITZ to the value 4.0.

EM as an INTEGER input variable is the number of eigenvalues to be computed, 0 .LT. EM .LT. P .LE. MN.

EM as an INTEGER output variable is the number of eigenvectors computed through KM iteration steps.

X as a real N-by-P input array is a set of P optional initial approximate eigenvectors X(I,1), ... , X(I,P), I = 1, ... , N, interpreted by SIMITZ if KM is negative.

X as a real N-by-P output array is a set of EM eigenvectors X(I,1), ... , X(I,EM), I = 1, ... , N, computed through IABS(KM) iteration steps with the remainder of X consisting of P - EM approximate eigenvectors. The N-by-P matrix X satisfies $X^TBX = I$, that is, the eigenvectors of C are B-orthonormal.

MN is an integer input variable which identifies the leading dimension in the calling program of the array X.

D is a real output P-array of which D(1), ... , D(EM) are the eigenvalues of C largest in magnitude in decreasing order corresponding to the computed eigenvectors X(I,1), ... , X(I,EM), I = 1, ... , N. D(EM+1), ... , D(P-1) contain approximations to progressively smaller such eigenvalues. D(P) contains the most recently computed value of E, where the interval (-E,E) is the interval over which the Chebyshev acceleration was performed.

WK the initial location of at least $P^2 + 3P + 2N = K$ consecutive storage locations which may not be overwritten while SIMITZ is

in execution.

SIMITZ employs a DATA statement to assign to a machine dependent REAL variable MT the quotient of the smallest positive REAL value representable by FORTRAN and the smallest REAL value whose sum with 1.0 exceeds 1.0. The performance of SIMITZ is strongly dependent upon the choice of input parameters and upon the careful preparation of the subprograms IP and OP. The user should develop experience with SIMITZ on problems of moderate size before investing processor time on very large problems for which the procedure is ultimately intended. During its execution SIMITZ issues calls to the following subprograms

FUNCTION RANF

returns uniformly distributed random numbers on the open interval (0, 1) given any one argument of any type.

SUBROUTINE TRED2

is the EISPACK program which computes a Householder tridiagonal form of a real symmetric matrix.

SUBROUTINE IMTQL2

is the EISPACK program which computes the eigenvalues and orthonormal eigenvectors of a symmetric tridiagonal matrix.

FUNCTION IP

is described above.

SUBROUTINE OP

is described above.

SUBROUTINE INF

is described above.

The Function RANF furnished with the CDC FORTRAN Common Library Mathematical Routines is an excellent random number generator. However, no failures of SIMITZ have been noted even when RANF was replaced by a crude in-line congruence procedure. The user should use the USA Standard FORTRAN versions of TRED2 or IMTQL2 rather than the DOUBLE.PRECISION versions available for the IBM 360/370 compatible processors. The EISPACK subroutine TQL2 may replace IMTQL2 if desired.

SECTION V

ALGORITHM

A complete listing follows of the simultaneous iteration algorithm for generalized symmetric matrices implemented in USA Standard FORTRAN as SUBROUTINE SIMITZ. The program and its documentation are separately sequenced.

```
      SUBROUTINE SIMITZ(N, P, KM, EPS, IP, OP, INF, EM, X, MN, D, WK)    SIMITZ    2
C*********************************************************************** SIMITZ/D  2
C       IDENTIFICATION                                                    SIMITZ/D  3
C         SIMITZ - ITERATIVE COMPUTATION OF EIGENVALUES LARGEST IN MAGNI- SIMITZ/D  4
C                  TUDE AND CORRESPONDING EIGENVECTORS OF A REAL GENERAL- SIMITZ/D  5
C                  IZED SYMMETRIC MATRIX                                  SIMITZ/D  6
C         FORTRAN SUBROUTINE SUBPROGRAM                                   SIMITZ/D  7
C         US AIR FORCE FLIGHT DYNAMICS LABORATORY                         SIMITZ/D  8
C         WRIGHT-PATTERSON AFB, OHIO  45433                               SIMITZ/D  9
C       PURPOSE                                                           SIMITZ/D 10
C         A REAL N-SQUARE MATRIX C IS B-SYMMETRIC RELATIVE TO AN N-SQUARE SIMITZ/D 11
C         POSITIVE DEFINITE MATRIX B IN CASE BC = C'B WHERE C' IS THE     SIMITZ/D 12
C         TRANSPOSE OF C.  GIVEN AN OPTIONAL SET OF P INITIAL APPROXIMATE SIMITZ/D 13
C         EIGENVECTORS OF A REAL N-SQUARE B-SYMMETRIC MATRIX C CORRES-    SIMITZ/D 14
C         PONDING TO P EIGENVALUES OF C LARGEST IN MAGNITUDE, SIMITZ COM- SIMITZ/D 15
C         PUTES EM EIGENVALUES AND EM CORRESPONDING EIGENVECTORS TO A     SIMITZ/D 16
C         PRECISION DEPENDENT ON THE STRUCTURE OF B AND C AND ON A GIVEN  SIMITZ/D 17
C         TOLERANCE EPS.  THE MATRIX B IS PRESENTED TO SIMITZ AS AN ALGO- SIMITZ/D 18
C         RITHM FOR CALCULATING THE STANDARD INNER PRODUCT (W, BZ) = W'BZ SIMITZ/D 19
C         GIVEN COLUMN N-VECTORS W AND Z IMPLEMENTED AS A FORTRAN COM-    SIMITZ/D 20
C         PATIBLE REAL FUNCTION SUBPROGRAM.  THE MATRIX C IS PRESENTED AS SIMITZ/D 21
C         A SUBROUTINE SUBPROGRAM WHICH GIVEN A COLUMN N-VECTOR Z CALCU-  SIMITZ/D 22
C         LATES ITS IMAGE W = CZ UNDER THE B-SYMMETRIC MATRIX C.  DEPEND- SIMITZ/D 23
C         ING ON THE CHOICE OF B AND C, SIMITZ APPLIES TO A WIDE VARIETY  SIMITZ/D 24
C         OF SYMMETRIC EIGENPROBLEMS.                                     SIMITZ/D 25
C       CONTROL                                                           SIMITZ/D 26
C                                                                         SIMITZ/D 27
C         DIMENSION X(MN,P), D(P), WK(K)                                  SIMITZ/D 28
C         INTEGER P, EM                                                   SIMITZ/D 29
C         REAL IP                                                         SIMITZ/D 30
C         EXTERNAL IP, INF, OP                                            SIMITZ/D 31
C         .                                                               SIMITZ/D 32
C         .                                                               SIMITZ/D 33
C         .                                                               SIMITZ/D 34
C         CALL SIMITZ(N, P, KM, EPS, IP, OP, INF, EM, X, MN, D, WK)       SIMITZ/D 35
C                                                                         SIMITZ/D 36
C         WHERE                                                           SIMITZ/D 37
C         N    IS AN INTEGER INPUT VARIABLE, THE ORDER OF THE MATRIX C.   SIMITZ/D 38
C         P    IS AN INTEGER INPUT VARIABLE, THE NUMBER OF SIMULTANEOUS   SIMITZ/D 39
C              ITERATION VECTORS.                                         SIMITZ/D 40
C         KM   AS AN INTEGER INPUT VARIABLE IS IN MAGNITUDE THE MAXIMUM   SIMITZ/D 41
C              NUMBER OF ITERATION STEPS TO BE EXECUTED.  IF KM IDENTIFIES SIMITZ/D 42
C              A NEGATIVE VALUE THEN P INITIAL APPROXIMATE EIGENVECTORS   SIMITZ/D 43
C              ARE ASSUMED TO BE PRESENT IN THE ARRAY X.  OTHERWISE SIMITZ SIMITZ/D 44
C              SUPPLIES RANDOM INITIAL EIGENVECTORS.                      SIMITZ/D 45
C         KM   AS AN INTEGER OUTPUT VARIABLE IDENTIFIES THE NUMBER KS OF  SIMITZ/D 46
C              ITERATION STEPS FINALLY USED IN THE CALCULATION OF EM      SIMITZ/D 47
C              EIGENVECTORS.                                              SIMITZ/D 48
C         EPS  IS A REAL INPUT VARIABLE, THE TOLERANCE FOR ACCEPTING      SIMITZ/D 49
C              EIGENVECTORS.                                              SIMITZ/D 50
C         IP   IS AN EXTERNAL INPUT VARIABLE, THE NAME OF A FORTRAN COM-  SIMITZ/D 51
C              PATIBLE REAL FUNCTION SUBPROGRAM OF THE FORM IP(N, Z, W)   SIMITZ/D 52
C              WHICH MUST RETURN THE INNER PRODUCT (W, BZ) = W'BZ OF THE  SIMITZ/D 53
C              VECTORS IDENTIFIED BY THE N-ARRAYS Z AND W RELATIVE TO THE SIMITZ/D 54
C              POSITIVE DEFINITE MATRIX B.                                SIMITZ/D 55
C         OP   IS AN EXTERNAL INPUT VARIABLE, THE NAME OF A FORTRAN COM-  SIMITZ/D 56
C              PATIBLE SUBROUTINE SUBPROGRAM OF THE FORM OP(N, Z, W)      SIMITZ/D 57
```

```
C            WHICH MUST CALCULATE THE IMAGE W OF THE VECTOR IDENTIFIED   SIMITZ/D 58
C            BY THE N-ARRAY Z UNDER THE N-SQUARE MATRIX C WITHOUT OVER-  SIMITZ/D 59
C            WRITING Z.                                                  SIMITZ/D 60
C         INF IS AN EXTERNAL INPUT VARIABLE, THE NAME OF A FORTRAN COM-  SIMITZ/D 61
C            PATIBLE SUBROUTINE SUBPROGRAM WHICH MAY BE USED FOR         SIMITZ/D 62
C            OBTAINING INFORMATION OR TO EXERT CONTROL DURING EXECUTION  SIMITZ/D 63
C            OF SIMITZ.  INF HAS THE FORM INF(KS, G, H, F) WHERE         SIMITZ/D 64
C              KS IS AN INTEGER OUTPUT VARIABLE, THE NUMBER OF THE NEXT  SIMITZ/D 65
C                 ITERATION STEP.                                        SIMITZ/D 66
C              G  IS AN INTEGER OUTPUT VARIABLE, THE NUMBER OF ALREADY   SIMITZ/D 67
C                 ACCEPTED EIGENVECTORS.                                 SIMITZ/D 68
C              H  IS AN INTEGER OUTPUT VARIABLE, THE NUMBER OF ALREADY   SIMITZ/D 69
C                 ACCEPTED EIGENVALUES.                                  SIMITZ/D 70
C              F  IS A REAL OUTPUT VARIABLE P-ARRAY, ERROR QUANTITIES    SIMITZ/D 71
C                 MEASURING RESPECTIVELY THE STATE OF CONVERGENCE OF     SIMITZ/D 72
C                 THE P SIMULTANEOUS ITERATION VECTORS.  IN ADDITION,    SIMITZ/D 73
C                 IF CONVERGENCE FAILS IN SUBROUTINE IMTQL2 AFTER G      SIMITZ/D 74
C                 EIGENVECTORS HAVE BEEN ACCEPTED, THEN F(G+1) IS RE-    SIMITZ/D 75
C                 PLACED BY 100.*FLOAT(IERR) WHERE IERR IS THE ERROR     SIMITZ/D 76
C                 INDICATOR OUTPUT BY IMTQL2.  EACH ELEMENT OF THE ARRAY SIMITZ/D 77
C                 F IS INITIALLY SET BY SIMITZ TO THE VALUE 4.0.         SIMITZ/D 78
C         EM  AS AN INTEGER INPUT VARIABLE IS THE NUMBER OF EIGENVALUES  SIMITZ/D 79
C            TO BE COMPUTED, 0 .LT. EM .LT. P .LE. N .LE. MN.            SIMITZ/D 80
C         EM  AS AN INTEGER OUTPUT VARIABLE IS THE NUMBER OF EIGENVECTORS SIMITZ/D 81
C            COMPUTED THROUGH KM ITERATION STEPS.                        SIMITZ/D 82
C         X   AS A REAL N-BY-P INPUT ARRAY IS A SET OF P OPTIONAL INITIAL SIMITZ/D 83
C            APPROXIMATE EIGENVECTORS X(I,1), ..., X(I,P), I = 1, ...,   SIMITZ/D 84
C            N, INTERPRETED BY SIMITZ IF KM IS NEGATIVE.                 SIMITZ/D 85
C         X   AS A REAL N-BY-P OUTPUT ARRAY IS A SET OF EM EIGENVECTORS  SIMITZ/D 86
C            X(I,1), ..., X(I,EM), I = 1, ..., N, COMPUTED THROUGH       SIMITZ/D 87
C            IABS(KM) ITERATION STEPS WITH THE REMAINDER OF X CONSISTING SIMITZ/D 88
C            OF P - EM APPROXIMATE EIGENVECTORS.  THE N-BY-P MATRIX X    SIMITZ/D 89
C            SATISFIES X'BX = I, THAT IS, THE EIGENVECTORS OF C ARE      SIMITZ/D 90
C            B-ORTHOGONAL.                                               SIMITZ/D 91
C         MN  IS AN INTEGER INPUT VARIABLE WHICH IDENTIFIES THE LEADING  SIMITZ/D 92
C            DIMENSION IN THE CALLING PROGRAM OF THE ARRAY X.            SIMITZ/D 93
C         D   IS A REAL OUTPUT P-ARRAY OF WHICH D(1), ..., D(EM) ARE THE SIMITZ/D 94
C            EIGENVALUES OF C LARGEST IN MAGNITUDE IN DECREASING ORDER   SIMITZ/D 95
C            CORRESPONDING TO THE COMPUTED EIGENVECTORS X(I,1), ...,     SIMITZ/D 96
C            X(I,EM), I = 1, ..., N.  D(EM+1), ..., D(P-1) CONTAIN       SIMITZ/D 97
C            APPROXIMATIONS TO PROGRESSIVELY SMALLER SUCH EIGENVALUES.   SIMITZ/D 98
C            D(P) CONTAINS THE MOST RECENTLY COMPUTED VALUE OF E, WHERE   SIMITZ/D 99
C            THE INTERVAL (-E, E) IS THE INTERVAL OVER WHICH THE         SIMITZ/D100
C            CHEBYSHEV ACCELERATION WAS PERFORMED.                       SIMITZ/D101
C         WK  THE INITIAL LOCATION OF AT LEAST P**2 + 3*P + 2*N = K      SIMITZ/D102
C            CONSECUTIVE STORAGE LOCATIONS WHICH MAY NOT BE OVER-        SIMITZ/D103
C            WRITTEN WHILE SIMITZ IS IN EXECUTION.                       SIMITZ/D104
C      OTHER PROGRAMMING INFORMATION                                     SIMITZ/D105
C         SIMITZ EMPLOYS A DATA STATEMENT TO ASSIGN TO A MACHINE DEPEN-  SIMITZ/D106
C         DENT REAL VARIABLE MT THE QUOTIENT OF THE SMALLEST POSITIVE    SIMITZ/D107
C         REAL VALUE REPRESENTABLE BY FORTRAN AND THE SMALLEST REAL VALUE SIMITZ/D108
C         WHOSE SUM WITH 1.0 EXCEEDS 1.0.                               SIMITZ/D109
C                                                                       SIMITZ/D110
C         THE PERFORMANCE OF SIMITZ IS STRONGLY DEPENDENT UPON THE CHOICE SIMITZ/D111
C         OF INPUT PARAMETERS AND UPON THE CAREFUL PREPARATION OF THE    SIMITZ/D112
C         SUBPROGRAMS IP AND OP.  THE USER SHOULD DEVELOP EXPERIENCE     SIMITZ/D113
C         WITH SIMITZ ON PROBLEMS OF MODERATE SIZE BEFORE INVESTING      SIMITZ/D114
```

```
C         PROCESSOR TIME ON VERY LARGE PROBLEMS FOR WHICH THE PROCEDURE   SIMITZ/D115
C         IS ULTIMATELY INTENDED.                                         SIMITZ/D116
C       OTHER PROGRAMS REQUIRED                                           SIMITZ/D117
C         FUNCTION RANF                                                   SIMITZ/D118
C             RETURNS UNIFORMLY DISTRIBUTED RANDOM NUMBERS ON THE OPEN    SIMITZ/D119
C             INTERVAL (0, 1) GIVEN ANY ONE ARGUMENT OF ANY TYPE.         SIMITZ/D120
C         SUBROUTINE TRED2                                                SIMITZ/D121
C             IS THE EISPACK (4) PROGRAM WHICH COMPUTES A HOUSEHOLDER     SIMITZ/D122
C             TRIDIAGONAL FORM OF A REAL SYMMETRIC MATRIX.                SIMITZ/D123
C         SUBROUTINE IMTQL2                                               SIMITZ/D124
C             IS THE EISPACK PROGRAM WHICH COMPUTES THE EIGENVALUES AND   SIMITZ/D125
C             ORTHONORMAL EIGENVECTORS OF A SYMMETRIC TRIDIAGONAL MATRIX. SIMITZ/D126
C         FUNCTION IP                                                     SIMITZ/D127
C             IS DESCRIBED ABOVE.                                         SIMITZ/D128
C         SUBROUTINE OP                                                   SIMITZ/D129
C             IS DESCRIBED ABOVE.                                         SIMITZ/D130
C         SUBROUTINE INF                                                  SIMITZ/D131
C             IS DESCRIBED ABOVE.                                         SIMITZ/D132
C       METHOD                                                            SIMITZ/D133
C         SIMITZ REPRESENTS RESULTS OF EXTENSIVE MODIFICATIONS AND TESTS  SIMITZ/D134
C         OF SUBROUTINE RITZIT (1), AN ANSI FORTRAN TRANSLATION OF THE    SIMITZ/D135
C         ALGOL 60 PROCEDURE OF THE SAME NAME (3).  THE BASIC RUTISHAUSER SIMITZ/D136
C         -REINSCH ALGORITHM IS PRESERVED.                                SIMITZ/D137
C       REFERENCES                                                        SIMITZ/D138
C         (1) PAUL J. NIKOLAI AND NAI-KUAN TSAO, THE AFL LINEAR ALGEBRA   SIMITZ/D139
C             LIBRARY HANDBOOK, AFL TR 74-0106, AEROSPACE RESEARCH LABOR- SIMITZ/D140
C             ATORIES, WRIGHT-PATTERSON AFB, OHIO, 1974.                  SIMITZ/D141
C         (2) HEINZ RUTISHAUSER, COMPUTATIONAL ASPECTS OF F.L. BAUER'S    SIMITZ/D142
C             SIMULTANEOUS ITERATION METHOD, NUMER. MATH. 13(1969), 4-13. SIMITZ/D143
C         (3) ----------------, SIMULTANEOUS ITERATION METHOD FOR SYM-     SIMITZ/D144
C             METRIC MATRICES, NUMER. MATH. 16(1970), 205-223.            SIMITZ/D145
C         (4) B.T. SMITH ET AL, MATRIX EIGENSYSTEM ROUTINES-EISPACK       SIMITZ/D146
C             GUIDE, LECTURE NOTES IN COMPUTER SCIENCE 6, SPRINGER-VERLAG SIMITZ/D147
C             NEW YORK, 1974.                                             SIMITZ/D148
C*********************************************************************** SIMITZ/D149
      EXTERNAL    INF,     IP,     OP                                     SIMITZ    4
      INTEGER      EM,      G,      H,      I,     IG,     IK,      J,    SIMITZ    5
     *             JK,     JP,      K,     KM,     KS,      L,     LF,    SIMITZ    6
     *             L1,      M,     MN,     M1,      N,      P,     Z1,    SIMITZ    7
     *             Z2                                                     SIMITZ    8
      LOGICAL    ORIG                                                     SIMITZ    9
      REAL          D,      E,    EPS,     E1,     E2,     IP,     MT,   SIMITZ   10
     *              S,      T,     WK,      X                             SIMITZ   11
      DIMENSION X(MN,1), D(1), WK(P,P,1)                                  SIMITZ   12
C                                                                         SIMITZ   13
      DATA MT /.2203606415850628-279/                                     SIMITZ   14
C                                                                         SIMITZ   15
C         THE LOCAL VARIABLE ARRAYS FROM (3) ARE ASSIGNED TO THE          SIMITZ   16
C                  VARIABLE ARRAY WK AS FOLLOWS:                          SIMITZ   17
C                                                                         SIMITZ   18
C         WK(I,J,1) = B(I,J), I, J = 1, ..., P.                          SIMITZ   19
C         WK(I,1,2) = QX(I), I = 1, ..., P.                               SIMITZ   20
C         WK(I,2,2) = F(I), I = 1, ..., P.                                SIMITZ   21
C         WK(I,3,2) = RQ(I), I = 1, ..., P.                               SIMITZ   22
C         WK(I,4,2) = U(I), I = 1, ..., N.                                SIMITZ   23
C         WK(I+N,4,2) = W(I), I = 1, ..., N.                              SIMITZ   24
C         NOT NEEDED ARE V(I), I = 1, ..., N, R(I), I = 1, ..., P, AND    SIMITZ   25
```

```
C         D(I,J), I, J = 1, ..., P.                                    SIMITZ   26
C                                                                      SIMITZ   27
C                    THE NEXT STATEMENT IS START.                      SIMITZ   28
      E = .0E+00                                                       SIMITZ   29
      G = 0                                                            SIMITZ   30
      IG = 1                                                           SIMITZ   31
      H = 0                                                            SIMITZ   32
      Z1 = 0                                                           SIMITZ   33
      Z2 = 0                                                           SIMITZ   34
      KS = 0                                                           SIMITZ   35
      M = 1                                                            SIMITZ   36
      WK(P,1,2) = .0E+00                                               SIMITZ   37
      DO 10 L = 1, P                                                   SIMITZ   38
        WK(L,2,2) = .4E+01                                             SIMITZ   39
        WK(L,3,2) = .0E+00                                             SIMITZ   40
 10   CONTINUE                                                         SIMITZ   41
      IF (KM) 50, 50, 20                                               SIMITZ   42
 20   DO 40 L = 1, P                                                   SIMITZ   43
        DO 30 J = 1, N                                                 SIMITZ   44
          X(J,L) = .2E+01*RANF(.2E+01) - .1E+01                        SIMITZ   45
 30     CONTINUE                                                       SIMITZ   46
 40   CONTINUE                                                         SIMITZ   47
 50   KM = IABS(KM)                                                    SIMITZ   48
      ASSIGN 60 TO IK                                                  SIMITZ   49
      LF = IG                                                          SIMITZ   50
      L1 = P                                                           SIMITZ   51
      GO TO 990                                                        SIMITZ   52
C         RAYLEIGH-RITZ STEP                                           SIMITZ   53
C                    STATEMENT 60 IS LOOP.                             SIMITZ   54
 60   DO 80 K = IG, P                                                  SIMITZ   55
        CALL OP(N, X(1,K), WK(1,4,2))                                  SIMITZ   56
        DO 70 J = 1, N                                                 SIMITZ   57
          X(J,K) = WK(J,4,2)                                           SIMITZ   58
 70     CONTINUE                                                       SIMITZ   59
 80   CONTINUE                                                         SIMITZ   60
      ASSIGN 90 TO IK                                                  SIMITZ   61
      LF = IG                                                          SIMITZ   62
      L1 = P                                                           SIMITZ   63
      GO TO 990                                                        SIMITZ   64
 90   IF (KS) 150, 100, 150                                            SIMITZ   65
C         MEASURES AGAINST UNHAPPY CHOICE OF INITIAL VECTORS           SIMITZ   66
 100  DO 130 K = 1, P                                                  SIMITZ   67
        IF (WK(K,K,1)) 130, 110, 130                                   SIMITZ   68
 110    DO 120 I = 1, N                                                SIMITZ   69
          X(I,K) = .2E+01*RANF(.2E+01) - .1E+01                        SIMITZ   70
 120    CONTINUE                                                       SIMITZ   71
        KS = 1                                                         SIMITZ   72
 130  CONTINUE                                                         SIMITZ   73
      IF (KS - 1) 150, 140, 150                                        SIMITZ   74
 140  ASSIGN 60 TO IK                                                  SIMITZ   75
      LF = 1                                                           SIMITZ   76
      L1 = P                                                           SIMITZ   77
      GO TO 990                                                        SIMITZ   78
 150  DO 180 K = IG, P                                                 SIMITZ   79
        DO 170 L = K, P                                                SIMITZ   80
          S = .0E+00                                                   SIMITZ   81
          DO 160 I = L, P                                              SIMITZ   82
```

```
            S = S + WK(I,K,1)*WK(I,L,1)                                    SIMITZ   83
  160       CONTINUE                                                       SIMITZ   84
            WK(L,K,1) = -S                                                 SIMITZ   85
  170     CONTINUE                                                         SIMITZ   86
  180   CONTINUE                                                           SIMITZ   87
        CALL TRED2(P, P - G, WK(IG,IG,1), D(IG), WK(1,4,2), WK(IG,IG,1))   SIMITZ   88
        CALL IMTQL2(P, P - G, D(IG), WK(1,4,2), WK(IG,IG,1), L)            SIMITZ   89
        WK(IG,2,2) = AMAX1(WK(IG,2,2), .1E+04*FLOAT(L))                    SIMITZ   90
        DO 190 K = IG, P                                                   SIMITZ   91
          D(K) = SQRT(AMAX1(-D(K), .0E+00))                                SIMITZ   92
  190   CONTINUE                                                           SIMITZ   93
C         REORDERING EIGENVALUES AND EIGENVECTORS ACCORDING TO SIZE OF     SIMITZ   94
C         THE FORMER IS ACCOMPLISHED IN SUBROUTINE IMTQL2.                 SIMITZ   95
        DO 230 J = 1, N                                                    SIMITZ   96
          DO 210 K = IG, P                                                 SIMITZ   97
            S = .0E+00                                                     SIMITZ   98
            DO 200 L = IG, P                                               SIMITZ   99
              S = S + X(J,L)*WK(L,K,1)                                     SIMITZ  100
  200       CONTINUE                                                       SIMITZ  101
            WK(K,4,2) = S                                                  SIMITZ  102
  210     CONTINUE                                                         SIMITZ  103
          DO 220 K = IG, P                                                 SIMITZ  104
            X(J,K) = WK(K,4,2)                                             SIMITZ  105
  220     CONTINUE                                                         SIMITZ  106
  230   CONTINUE                                                           SIMITZ  107
        KS = KS + 1                                                        SIMITZ  108
        E = AMAX1(D(P), E)                                                 SIMITZ  109
C         RANDOMIZATION                                                    SIMITZ  110
        IF (3 - Z1) 260, 240, 240                                          SIMITZ  111
  240   DO 250 J = 1, N                                                    SIMITZ  112
          X(J,P) = .2E+01*RANF(.2E+01) - .1E+01                            SIMITZ  113
  250   CONTINUE                                                           SIMITZ  114
        JP = P - 1                                                         SIMITZ  115
        ASSIGN 260 TO IK                                                   SIMITZ  116
        LF = P                                                             SIMITZ  117
        L1 = P                                                             SIMITZ  118
        GO TO 990                                                          SIMITZ  119
C         COMPUTE CONTROL QUANTITIES CX(I).                                SIMITZ  120
  260   DO 310 K = IG, JP                                                  SIMITZ  121
          S = (D(K) - E)*(D(K) + E)                                        SIMITZ  122
          IF (S) 270, 270, 280                                             SIMITZ  123
  270     WK(K,1,2) = .0E+00                                               SIMITZ  124
          GO TO 310                                                        SIMITZ  125
  280     IF (E) 300, 290, 300                                             SIMITZ  126
  290     WK(K,1,2) = .1E+04 + ALOG(D(K))                                  SIMITZ  127
          GO TO 310                                                        SIMITZ  128
  300     WK(K,1,2) = ALOG((D(K) + SQRT(S))/E)                             SIMITZ  129
  310   CONTINUE                                                           SIMITZ  130
C         ACCEPTANCE TEST FOR EIGENVALUES INCLUDING ADJUSTMENT OF EM AND   SIMITZ  131
C         H SUCH THAT D(EM) .GT. E, D(H) .GT. E AND D(EM) DOES NOT         SIMITZ  132
C         OSCILLATE STRONGLY                                               SIMITZ  133
        I = Z1 - 1                                                         SIMITZ  134
        K = G                                                              SIMITZ  135
  320   K = K + 1                                                          SIMITZ  136
          IF (EM - K) 370, 330, 330                                        SIMITZ  137
  330     IF (D(K) - E) 360, 360, 340                                      SIMITZ  138
  340     IF (I) 320, 320, 350                                             SIMITZ  139
```

```
 350    IF (D(K) - .999E+00*WK(K,3,2)) 360, 360, 320                    SIMITZ  140
 360  CONTINUE                                                          SIMITZ  141
      EM = K - 1                                                        SIMITZ  142
C                 STATEMENT 370 IS EX4.                                 SIMITZ  143
 370  IF (EM) 380, 1130, 380                                            SIMITZ  144
 380  K = H                                                             SIMITZ  145
      S = .1E+01 + .1E+00*EPS                                           SIMITZ  146
 390  K = K + 1                                                         SIMITZ  147
        IF (D(K)) 400, 410, 400                                         SIMITZ  148
 400    IF (D(K) - S*WK(K,3,2)) 390, 390, 410                           SIMITZ  149
 410  CONTINUE                                                          SIMITZ  150
      H = K - 1                                                         SIMITZ  151
      K = EM                                                            SIMITZ  152
 420  K = K + 1                                                         SIMITZ  153
        IF (K - H) 430, 430, 450                                        SIMITZ  154
 430    IF (D(K) - E) 440, 440, 420                                     SIMITZ  155
 440  CONTINUE                                                          SIMITZ  156
      H = K - 1                                                         SIMITZ  157
C       ACCEPTANCE TEST FOR EIGENVECTORS                                SIMITZ  158
 450  L = G                                                             SIMITZ  159
      E2 = .0E+00                                                       SIMITZ  160
      DO 590 K = IG, JP                                                 SIMITZ  161
        IF (K - (L + 1)) 510, 460, 510                                  SIMITZ  162
C       CHECK FOR NESTED EIGENVALUES                                    SIMITZ  163
 460    L = K                                                           SIMITZ  164
        L1 = K                                                          SIMITZ  165
        S = .5E+00/FLOAT(KS)                                            SIMITZ  166
        T = .1E+01/FLOAT(KS*M)                                          SIMITZ  167
 470    L = L + 1                                                       SIMITZ  168
          IF (L - JP) 480, 480, 490                                     SIMITZ  169
 480      IF (WK(L,1,2)*(WK(L,1,2) + S) + T - WK(L-1,1,2)*WK(L-1,1,2))  SIMITZ  170
     *         490, 490, 470                                            SIMITZ  171
 490    CONTINUE                                                        SIMITZ  172
        L = L - 1                                                       SIMITZ  173
C                 THE NEXT STATEMENT IS EX5.                            SIMITZ  174
        IF (L - H) 510, 510, 500                                        SIMITZ  175
 500    L = L1 - 1                                                      SIMITZ  176
        GO TO 500                                                       SIMITZ  177
 510    CALL OP(N, X(1,K), WK(1,4,2))                                   SIMITZ  178
        S = .0E+00                                                      SIMITZ  179
        DO 540 J = 1, L                                                 SIMITZ  180
          IF (ABS(D(J) - D(K)) - .1E-01*D(K)) 520, 540, 540             SIMITZ  181
 520      T = IP(N, WK(1,4,2), X(1,J))                                  SIMITZ  182
          DO 530 I = 1, N                                               SIMITZ  183
            WK(I,4,2) = WK(I,4,2) - T*X(I,J)                            SIMITZ  184
 530      CONTINUE                                                      SIMITZ  185
          S = S + T*T                                                   SIMITZ  186
 540    CONTINUE                                                        SIMITZ  187
        T = .1E+01                                                      SIMITZ  188
        IF (S .NE. .0E+00) T = IP(N, WK(1,4,2), WK(1,4,2))              SIMITZ  189
        E2 = AMAX1(E2, SQRT(T/(S + T)))                                 SIMITZ  190
        IF (K - L) 590, 550, 590                                        SIMITZ  191
C       TEST FOR ACCEPTANCE OF GROUP OF EIGENVECTORS                    SIMITZ  192
 550    IF (L .GE. EM .AND. D(EM)*WK(EM,2,2) .LT. EPS*(D(EM) - E))      SIMITZ  193
     *    G = EM                                                       SIMITZ  194
        IF (E2 - WK(L,2,2)) 560, 580, 580                               SIMITZ  195
 560    DO 570 J = L1, L                                                SIMITZ  196
```

```
          WK(J,2,2) = E2                                                    SIMITZ  197
  570     CONTINUE                                                          SIMITZ  198
  580     IF (L .LE. EM .AND. D(L)*WK(L,2,2) .LT. EPS*(D(L) - E)) G = L     SIMITZ  199
  590   CONTINUE                                                            SIMITZ  200
C         ADJUST M.                                                         SIMITZ  201
C                 STATEMENT 600 IS EX6.                                     SIMITZ  202
  600   IG = G + 1                                                          SIMITZ  203
        IF (E - .4E-01*D(1)) 610, 610, 620                                  SIMITZ  204
  610   M = 1                                                               SIMITZ  205
        K = 1                                                               SIMITZ  206
        GO TO 630                                                           SIMITZ  207
  620   E2 = .2E+01/E                                                       SIMITZ  208
        E1 = .51E+00*E2                                                     SIMITZ  209
        K = 2*INT(.4E+01/AMIN1(WK(1,1,2), .4E+01))                          SIMITZ  210
        M = MIN0(M, K)                                                      SIMITZ  211
C         REDUCE EM IF CONVERGENCE WOULD BE TOO SLOW.                       SIMITZ  212
  630   IF (WK(EM,2,2)) 640, 690, 640                                       SIMITZ  213
  640   IF (FLOAT(KS) - .9E+00*FLOAT(KM)) 650, 690, 690                     SIMITZ  214
  650   S = FLOAT(K)*WK(EM,1,2)                                             SIMITZ  215
        IF (S - .5E-01) 660, 670, 670                                       SIMITZ  216
  660   T = .5E+00*S*WK(EM,1,2)                                             SIMITZ  217
        GO TO 680                                                           SIMITZ  218
  670   T = WK(EM,1,2) + ALOG(.5E+00 + .5E+00*EXP(-.2E+01*S))/FLOAT(K)      SIMITZ  219
  680   S = ALOG(D(EM)*WK(EM,2,2)/(EPS*(D(EM) - E)))                        SIMITZ  220
        IF (S*FLOAT(KS) .GT. T*FLOAT((KM - KS)*KM)) EM = EM - 1             SIMITZ  221
C                 STATEMENT 690 IS EX2.                                     SIMITZ  222
  690   DO 700 K = IG, JP                                                   SIMITZ  223
          WK(K,3,2) = D(K)                                                  SIMITZ  224
  700   CONTINUE                                                            SIMITZ  225
        CALL INF(KS, G, H, WK(1,2,2))                                       SIMITZ  226
        IF (G .GE. EM .OR. KS .GE. KM) GO TO 1130                           SIMITZ  227
C                 STATEMENT 710 IS EX1.                                     SIMITZ  228
  710   IF (KS + M - KM) 730, 730, 720                                      SIMITZ  229
  720   Z2 = -1                                                             SIMITZ  230
        IF (M .GT. 1) M = 2*((KM - KS + 1)/2)                               SIMITZ  231
  730   M1 = M                                                              SIMITZ  232
C         SHORTCUT LAST INTERMEDIATE BLOCK IF ALL F(I) ARE SUFFICIENTLY     SIMITZ  233
C         SMALL.                                                            SIMITZ  234
        IF (L - EM) 780, 740, 740                                           SIMITZ  235
  740   S = D(EM)*WK(EM,2,2)/(EPS*(D(EM) - E))                              SIMITZ  236
        T = S*S - .1E+01                                                    SIMITZ  237
        IF (T) 60, 60, 750                                                  SIMITZ  238
  750   S = ALOG(S + SQRT(T))/(WK(EM,1,2) - WK(H+1,1,2))                    SIMITZ  239
        M1 = 2*INT(.5E+00*S + .101E+01)                                     SIMITZ  240
        IF (M1 - M) 770, 770, 760                                           SIMITZ  241
  760   M1 = M                                                              SIMITZ  242
        GO TO 780                                                           SIMITZ  243
  770   Z2 = -1                                                             SIMITZ  244
C         CHEBYSHEV ITERATION                                               SIMITZ  245
  780   IF (M - 1) 900, 790, 820                                            SIMITZ  246
  790   DO 810 K = IG, P                                                    SIMITZ  247
          CALL OP(N, X(1,K), WK(1,4,2))                                     SIMITZ  248
          DO 800 I = 1, N                                                   SIMITZ  249
            X(I,K) = WK(I,4,2)                                              SIMITZ  250
  800     CONTINUE                                                          SIMITZ  251
  810   CONTINUE                                                            SIMITZ  252
        GO TO 900                                                           SIMITZ  253
```

```
  820  L1 = M1 - 4                                                          SIMIT7  254
       DO 890 K = IG, P                                                     SIMIT7  255
         CALL OP(N, X(1,K), WK(1,4,2))                                      SIMIT7  256
         DO 830 I = 1, N                                                    SIMIT7  257
           IK = I + N                                                       SIMIT7  258
           WK(IK,4,2) = E1*WK(I,4,2)                                        SIMIT7  259
  830    CONTINUE                                                           SIMIT7  26.
         CALL OP(N, WK(N+1,4,2), WK(1,4,2))                                 SIMIT7  261
         DO 840 I = 1, N                                                    SIMIT7  262
           X(I,K) = E2*WK(I,4,2) - X(I,K)                                   SIMIT7  263
  840    CONTINUE                                                           SIMIT7  264
         IF (L1) 890, 850, 850                                              SIMIT7  265
  850    DO 880 J = 4, M1, 2                                                SIMIT7  266
           CALL OP(N, X(1,K), WK(1,4,2))                                    SIMIT7  267
           DO 860 I = 1, N                                                  SIMIT7  268
             IK = I + N                                                     SIMIT7  269
             WK(IK,4,2) = E2*WK(I,4,2) - WK(IK,4,2)                         SIMIT7  27.
  860      CONTINUE                                                         SIMIT7  271
           CALL OP(N, WK(N+1,4,2), WK(1,4,2))                               SIMIT7  272
           DO 870 I = 1, N                                                  SIMIT7  273
             X(I,K) = E2*WK(I,4,2) - X(I,K)                                 SIMIT7  274
  870      CONTINUE                                                         SIMIT7  275
  880    CONTINUE                                                           SIMIT7  276
  890  CONTINUE                                                             SIMIT7  277
  900  ASSIGN 910 TO IK                                                     SIMIT7  278
       LF = IG                                                              SIMIT7  279
       L1 = P                                                               SIMIT7  28.
       GO TO 990                                                            SIMIT7  281
C        DISCOUNTING THE ERROR QUANTITIES F                                 SIMIT7  282
  910  IF (G - H) 920, 970, 970                                             SIMIT7  283
  920  IF (M - 1) 950, 930, 950                                             SIMIT7  284
  930  DO 940 K = IG, H                                                     SIMIT7  285
         WK(K,2,2) = WK(K,2,2)*(D(H+1)/D(K))                                SIMIT7  286
  940  CONTINUE                                                             SIMIT7  287
       GO TO 970                                                            SIMIT7  288
  950  T = EXP(-FLOAT(M1)*WK(H+1,1,2))                                      SIMIT7  289
       DO 960 K = IG, H                                                     SIMIT7  29.
         S = EXP(-FLOAT(M1)*(WK(K,1,2) - WK(H+1,1,2)))                      SIMIT7  291
         WK(K,2,2) = S*WK(K,2,2)*(.1E+01 + T*T)/(.1E+01 + (S*T)*(S*T))      SIMIT7  292
  960  CONTINUE                                                             SIMIT7  293
  970  KS = KS + M1                                                         SIMIT7  294
       Z2 = Z2 - M1                                                         SIMIT7  295
C        POSSIBLE REPETITION OF INTERMEDIATE STEPS                          SIMIT7  296
       IF (Z2) 980, 710, 710                                                SIMIT7  297
  980  Z1 = Z1 + 1                                                          SIMIT7  298
       Z2 = 2*Z1                                                            SIMIT7  299
       M = M + M                                                            SIMIT7  30.
       GO TO 60                                                             SIMIT7  301
C        PERFORMS ORTHONORMALIZATION OF COLUMNS 1 THROUGH L1 OF ARRAY       SIMIT7  302
C        X ASSUMING THAT COLUMNS 1 THROUGH LF - 1 ARE ALREADY ORTHO-        SIMIT7  303
C        NORMAL                                                             SIMIT7  304
  990  DO 1120 K = LF, L1                                                   SIMIT7  305
         ORIG = .TRUE.                                                      SIMIT7  306
 1000    T = .0E+00                                                         SIMIT7  307
         JK = K - 1                                                         SIMIT7  308
         IF (JK) 1040, 1040, 1010                                           SIMIT7  309
 1010    DO 1030 I = 1, JK                                                  SIMIT7  31.
```

```
          S = IP(N, X(1,I), X(1,K))                                    SIMITZ  311
          IF (ORIG) WK(K,I,1) = S                                      SIMITZ  312
          T = T + S*S                                                  SIMITZ  313
          DO 1020 J = 1, N                                             SIMITZ  314
            X(J,K) = X(J,K) - S*X(J,I)                                 SIMITZ  315
 1020     CONTINUE                                                     SIMITZ  316
 1030   CONTINUE                                                       SIMITZ  317
 1040   S = IP(N, X(1,K), X(1,K))                                      SIMITZ  318
        T = S + T                                                      SIMITZ  319
        IF (S - .1E-01*T) 1060, 1060, 1050                             SIMITZ  320
 1050   IF (T - MT) 1060, 1060, 1080                                   SIMITZ  321
 1060   ORIG = .FALSE.                                                 SIMITZ  322
        IF (S - MT) 1070, 1070, 1080                                   SIMITZ  323
 1070   S = .0E+00                                                     SIMITZ  324
 1080   S = SQRT(S)                                                    SIMITZ  325
        WK(K,K,1) = S                                                  SIMITZ  326
        IF (S) 1090, 1100, 1090                                        SIMITZ  327
 1090   S = .1E+01/S                                                   SIMITZ  328
 1100   DO 1110 J = 1, N                                               SIMITZ  329
          X(J,K) = S*X(J,K)                                            SIMITZ  330
 1110   CONTINUE                                                       SIMITZ  331
 1120 CONTINUE                                                         SIMITZ  332
      GO TO IK, (60, 90, 260, 910, 1140)                               SIMITZ  333
C                 STATEMENT 1130 IS EX.                                SIMITZ  334
 1130 EM = G                                                           SIMITZ  335
C       SOLVE EIGENVALUE PROBLEM OF PROJECTION OF MATRIX C.            SIMITZ  336
      ASSIGN 1140 TO IK                                                SIMITZ  337
      LF = 1                                                           SIMITZ  338
      L1 = JP                                                          SIMITZ  339
      GO TO 930                                                        SIMITZ  340
 1140 DO 1160 K = 1, JP                                                SIMITZ  341
        CALL OP(N, X(1,K), X(1,P))                                     SIMITZ  342
        DO 1150 I = 1, K                                               SIMITZ  343
          WK(K,I,1) = -IP(N, X(1,I), X(1,P))                           SIMITZ  344
 1150   CONTINUE                                                       SIMITZ  345
 1160 CONTINUE                                                         SIMITZ  346
      CALL TRED2(P, JP, WK, D, WK(1,4,2), WK)                          SIMITZ  347
      CALL IMTQL2(P, JP, D, WK(1,4,2), WK, L)                          SIMITZ  348
      WK(IG,2,2) = AMAX1(WK(IG,2,2), .1E+04*FLOAT(L))                  SIMITZ  349
C       ARRANGE EIGENVALUES IN ORDER OF DECREASING ABSOLUTE VALUE.     SIMITZ  350
      DO 1210 J = 1, JP                                                SIMITZ  351
        K = J                                                          SIMITZ  352
        DO 1170 I = J, JP                                              SIMITZ  353
          IF (ABS(D(I)) .GT. ABS(D(K))) K = I                          SIMITZ  354
 1170   CONTINUE                                                       SIMITZ  355
        IF (K - J) 1200, 1200, 1180                                    SIMITZ  356
 1180   T = D(K)                                                       SIMITZ  357
        D(K) = D(J)                                                    SIMITZ  358
        D(J) = T                                                       SIMITZ  359
        DO 1190 I = 1, JP                                              SIMITZ  360
          T = WK(I,K,1)                                                SIMITZ  361
          WK(I,K,1) = WK(I,J,1)                                        SIMITZ  362
          WK(I,J,1) = T                                                SIMITZ  363
 1190   CONTINUE                                                       SIMITZ  364
 1200   D(J) = -D(J)                                                   SIMITZ  365
 1210 CONTINUE                                                         SIMITZ  366
      DO 1250 J = 1, N                                                 SIMITZ  367
```

```
      DO 1230 I = 1, JP                                                 SIMIT7  368
        S = .0E+00                                                      SIMIT7  369
        DO 1220 K = 1, JP                                               SIMIT7  370
          S = S + X(J,K)*WK(K,I,1)                                      SIMIT7  371
 1220   CONTINUE                                                        SIMIT7  372
        WK(I,4,2) = S                                                   SIMIT7  373
 1230 CONTINUE                                                          SIMIT7  374
      DO 1240 I = 1, JP                                                 SIMIT7  375
        X(J,I) = WK(I,4,2)                                              SIMIT7  376
 1240 CONTINUE                                                          SIMIT7  377
 1250 CONTINUE                                                          SIMIT7  378
      KM = KS                                                           SIMIT7  379
      D(P) = E                                                          SIMIT7  380
      RETURN                                                            SIMIT7  381
      END                                                               SIMIT7  382
```

REFERENCES

1. Arnurius, Donald E., Private communication.

2. Bauer, F.L., Das Verfaheren der Treppeniteration und Verwandte Verfaheren zur Lösung algebraischer Eigenvertprobleme, Z. Angew. Math. Phys. 8, 214-235 (1957).

3. Garbow, Burton S., Private communication.

4. Gupta, K.K., Solution of eigenvalue problems by Sturm sequence method, Int. J. Numer. Meth. Eng. 4, 379-404 (1972).

5. McCormick, S.F., and T. Noe, Simultaneous iteration for the matrix eigenvalue problem, J. Lin. Alg. Appl., To appear.

6. Nikolai, Paul J., and Nai-kuan Tsao, The ARL Linear Algebra Library Handbook, ARL TR 74-0106, Aerospace Research Laboratories, Wright-Patterson AFB, Ohio, 1974.

7. Peters, G., and J.H. Wilkinson, $Ax = \lambda Bx$ and the generalized eigenproblem, SIAM J. Numer. Anal. 7, 479-492 (1970).

8. Rutishauser, H., Computational aspects of F.L. Bauer's simultaneous iteration method, Numer. Math. 13, 4-13 (1969).

9. Rutishauser, H., Simultaneous iteration method for symmetric matrices, Numer. Math. 16, 205-223 (1970).

10. Smith, B.T., et al, Matrix Eigensystem routines - EISPACK Guide, Lecture Notes in Computer Science 6, Springer-Verlag, N.Y., 1974.

11. Underwood, Richard Ray, An iterative block Lanczos method for the solution of large sparse symmetric eigenproblems, Dissertation, Stanford University, STAN-CS-75-496, 1975.

12. Wilkinson, J.H., and C. Reinsch, Linear Algebra, Handbook for Automatic Computation II, Springer-Verlag, N.Y., 1971.

☆ U. S. GOVERNMENT PRINTING OFFICE: 1977 — 757-001/295